Data clustering using a linear cellular automata-based algorithm

Javier de Lope a,b,⁎, Darío Maravall a

⁎ Computational Cognitive Robotics Group, Department of Artificial Intelligence, Universidad Politécnica de Madrid, Madrid, Spain
a Department of Applied Intelligent Systems, Universidad Politécnica de Madrid, Madrid, Spain

Available online 26 October 2012

ARTICLE INFO

Keywords:
Cellular automata
Machine learning
Pattern recognition
Data mining
Data clustering
Social segregation models
Ants clustering

ABSTRACT

In this paper we propose a novel data clustering algorithm based on the idea of considering the individual data items as cells belonging to a uni-dimensional cellular automaton. Our proposed algorithm combines insights into both social segregation models based on Cellular Automata Theory, where the data items themselves are able to move autonomously in lattices, and also from Ants Clustering algorithms, particularly in the idea of distributing at random the data items to be clustered in lattices. We also consider an automatic method for determining the number of clusters in the dataset by analyzing the intra-cluster variances. A series of experiments with both synthetic and real datasets are presented in order to study empirically the convergence and performance results. These experimental results are compared to the obtained by conventional clustering algorithms.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Clustering is a method of unsupervised learning used in different fields such as Machine Learning, Data Mining, and Pattern Recognition. It deals with the assignment of a set of data items into subsets or clusters in such a way that the data items in the same cluster are similar.

Our proposed algorithm is inspired on social segregation models based on Cellular Automata Theory [1]. In this line of thought Schelling [2] originally proposed models of segregation in which the population is composed of two well-differentiated types of individuals. Each individual cares about the neighbors. If the neighbors dissatisfied him, then the individual moves to the nearest point that meets his minimum demand, i.e. the nearest point at which a significant proportion of his neighbors satisfy his demands.

Hegselmann et al. [3] also present similar ideas based on the use of Cellular Automata Theory by identifying the individuals with cells in a toroidal rectangular lattice and the individuals’ choices with the states that the cells can evolve. Under this approach, social phenomena such as the emergence of order, the micro-macro relations and the social dynamics can be analytically understood with the help of Cellular Automata Theory.

Several data clustering algorithms based on the social collective behavior have been proposed [4,5] and also in other different tasks [6–8].

Beckers et al. [10] ran a classic experiment in which a group of mobile robots gather several randomly distributed objects and cluster them into separated piles. The agents’ coordination in those experiments was achieved through stigmergy, principle originally developed for the description of termite building behavior. The agents’ behavior is determined by an indirect communication between them, which is carried out sensing and modification of the agents’ local environment. This situated experiment inspired several successful data clustering algorithms [11–13].

In ants clustering algorithms, a group or colony of ant-like agents perform a clustering task in a toroidal 2D rectangular lattice or grid in which the N-dimensional data items to be clustered have been previously scattered at random with a single data item at each site. The individual data items randomly scattered in the rectangular lattice can be picked-up, transported and dropped by the ants-like agents. The two basic actions of picking-up and dropping a data item are performed in a probabilistic way by the agents. Generally speaking, the probability for an agent of picking-up a particular data item depends proportionally on the number of similar data items deposited in its neighborhood, so that the lower the number of similar neighbors, the higher the probability of picking-up a particular data item. Conversely, the higher the number of similar neighbors, the higher the probability for a loaded agent to drop a data item at an empty site in the toroidal rectangular grid.

In this paper we propose a novel data clustering algorithm based on the idea of considering the individual data items as mobile agents. Our proposed algorithm combines insights into both ants clustering algorithms—particularly in the idea of distributing at random the data items to be clustered toroidal discrete lattices—and also from social segregation models based on Cellular Automata Theory, in which the data items themselves,
like individuals in a social neighborhood, are able to move autonomously in the toroidal linear lattice, following a basic rule for staying at a particular site if a similarity with the neighbors’ condition holds or for migrating from a particular location in which the similarity with the neighbors’ condition does not hold.

Usually some parameters required during the cluster analysis such as the number of classes or thresholds of dissimilarity between the items have to be defined by the designer. In our case the association of each item to a particular cluster can be performed by analyzing the chainmap diagram. We propose an automatic method for solving the hard task of determining the number of clusters in the dataset by analyzing the intra-cluster variances.

The rest of the paper is organized as follows. First we briefly present the idea of transforming the initial multi-dimensional data into a uni-dimensional dataset or lattice able to be clustered by a linear cellular automaton. Then, we describe the working of a linear cellular automata as a clustering device, which is the main and most original contribution of the paper. Also, we specify the automatic procedure for obtaining the optimal number of clusters in a dataset. Afterwards, we present the experiments aimed at testing the proposed method and its comparison with other existing methods. We have used both synthetic data as well as standard benchmark datasets like the Iris dataset. Finally, the paper ends with the conclusions of these comparative experiments.

2. One-dimensional cellular automata-based clustering

In this paper we propose to employ one-dimensional discrete lattices of cells upon which cellular automata operate performing an unsupervised data clustering process. Each cell \( x_i(t) \) in the discrete lattice \( L \) is the ith cell at time \( t \) and is related to a particular data item, \( x_i \), in the dataset. The number of cells in the lattice, \( N \), is also the number of data items in the dataset. As usual for one-dimensional lattices we are assuming periodic boundary conditions, in which \( x_{N+1} \) is identified with \( x_1 \).

At initialization (\( t=0 \)), the data items \( x_i \) are randomly associated to the cells \( x_i(t=0) \). Then, a set of transition rules \( \phi_r \) for \( r=3, \ldots, R \) is iteratively applied on a range of \( r \) cells in the lattice \( L \) (\( r \) specifies the size of the neighborhood for the rule \( \phi_r \)). The number of rules, \( R \), is computed in terms of the number of cells in the lattice, \( N \), and the size of the greater cluster: the greater \( R \), the greater the clusters sizes can be. The transitions are local in both space and time: a cell evolves according a function of the current state of that cell and its neighboring cells.

It is considered that an iteration step is achieved when all the rules \( \phi_r \) are applied to each cell in the lattice \( L \). Notice that \( r \) must be greater or equal to 3, i.e. the minimum neighborhood considered has three cells.

The process finishes when the lattice \( L \) converges to a final state, in which the states of each cell \( x_i(t) \) are stationary and equivalent to the previous states \( x_i(t-1) \), for each \( i = 1, \ldots, N \). Two states \( x_i \) and \( x_j \) are equivalent when the associated data items \( x_i \) and \( x_j \) belong to the same cluster.

A generic rule \( \phi_r \) for a neighborhood of \( r \) cells can be written as follows:

\[
[\phi_r(t+1), \, x_i(t+1), \, x_{i+r-1}(t+1)] = \phi_r(x_i(t), \, x_{i+1}(t), \, x_{i+r-1}(t))
\]

(1)

where \( i \) is the initial or reference cell. Each rule only considers the reference cell, the contiguous cell to the reference one, and the cell referred by the range of rule \( r \). Thus, the rule \( \phi_r \) considers sequentially the cells \( x_1, x_2 \) and \( x_3 \) for \( i=1 \), the cells \( x_2, x_3 \) and \( x_4 \) for \( i=2 \), and so on; the generic rule \( \phi_r \) considers sequentially the cells \( x_1, x_2 \) and \( x_3 \) for \( i=1 \), the cells \( x_2, x_3 \) and \( x_4 \) for \( i=2 \), and so on.

As we are assuming periodic boundary conditions, a rule \( \phi_r \) is completely applied when \( x_{i+r-1} \) matches with \( x_1 \). Then, the algorithm increases the range of rule and applies the next rule \( \phi_{r+1} \) to the whole lattice \( L \) again.

The rule \( \phi_r \) computes the distance between the feature vectors of the data items associated to the reference cell \( x_i(t) \) and its contiguous cell \( x_{i+1}(t) \) and compares it to the distance between the feature vectors of the data items associated to the reference cell \( x_i(t) \) and the last cell in that neighborhood \( x_{i+r-1}(t) \). The data items associated to the cell \( x_{i+1}(t+1) \) will be the data items with the lesser distance. Formally:

\[
u_{i+1}(t+1) = \begin{cases} u_{i+1}(t) & \text{if } d(x_i, x_{i+1}) \leq d(x_i, x_{i+r-1}) \\ u_{i+r-1}(t) & \text{otherwise} \end{cases}
\]

(2)

Equally, the data items associated to the cell \( x_{i+r-1}(t+1) \) will be the data item with the greater distance:

\[
u_{i+r-1}(t+1) = \begin{cases} u_{i+r-1}(t) & \text{if } d(x_i, x_{i+r-1}) \leq d(x_i, x_{i+1}) \\ u_{i+1}(t) & \text{otherwise} \end{cases}
\]

(3)

Namely, these rules keep together data with similar values in the feature vectors. Fig. 1 depicts the states which the tape can evolve depending on the data items considered.

By increasing systematically the range \( r \) of the rules, the number of contiguous cells belonged to data items on the same cluster also increases. After a few iterations, the system converges to a final state where all the cells associated to data items of the same cluster are contiguous.

3. Determining the number and composition of clusters

The next stage is a general issue of every unsupervised clustering algorithm and it deals with determining the number of clusters and associating the items to each cluster.

Our clustering algorithm based on one-dimensional cellular automata is able to obtain a partially ordered lattice in which each pair of consecutive cells contain references to the most similar items in the original dataset.

The items to be associated to each cluster can be determined by analyzing the chainmap diagram which is created by computing all

\[
\begin{array}{cccc}
\cdots & v_i & v_{i+1} & \cdots \\
\cdots & d(x_i, x_{i+1}) \leq d(x_i, x_{i+r-1}) \\
\cdots & v_i & v_{i+1} & \cdots \\
\cdots & d(x_i, x_{i+1}) > d(x_i, x_{i+r-1}) \\
\cdots & v_i & v_{i+1} & \cdots
\end{array}
\]

Fig. 1. A tape state can evolve to two different states depending on the distances between data item associated to the reference cell \( v_i \) and the cells \( v_{i+1} \) and \( v_{i+r-1} \).
the distances between the data associated to each cell and the next one and maintaining the periodic boundary conditions. When a distance is much greater than the previous ones, the new cell corresponds to a new cluster. On the opposite, while the distance between the items belonged to two consecutive cells keeps low, it can be considered that those items correspond to the same cluster.

This procedure is reminded the basic sequential clustering algorithm [14] although in our proposal to re-analyze the chain of items again is not needed because we use the output of the cellular automaton. Moreover, any threshold of dissimilarity nor the maximum allowable number of clusters as in the conventional basic sequential clustering algorithm is not needed. We find the appropriate number of clusters by computing an index based on the intra-cluster variances. The number of clusters in the dataset corresponds to the minimum of this index for all the possible number of clusters. The index $J$ is defined as follows:

$$J = \left\| \sum_{i=1}^{c} \sigma_i \right\|$$

where $c$ is the number of clusters and $\sigma_i^2$ is the intra-cluster variance of the cluster $i$. As we are considering multivariate datasets, we use the variance of each variable separately and we compute the norm of the result vector in order to get the index $J$.

On some occasions some clusters could be composed just for one data item which can be easily detected. Usually these spare data items are very near to the centroid of a cluster. When it occurs, we introduce a refinement to the solution by associating the spare items to the nearest clusters and reducing the number of classes.

4. Experimental results

We have tested the proposed clustering algorithm based on the cellular automata as the automatic procedure for determining the number of clusters with both synthetic and real data like the well-known Iris dataset. In the sequel we describe and comment the results.

4.1. Clustering results with synthetic data

Fig. 2 depicts the synthetic dataset used for experimental purposes. It is composed of four two-dimensional classes. Each class contains 20 data items.

After initializing the linear lattice by depositing at random the individual data items, one at each cell or site, the cellular automaton rules explained in Section 2 start to act. Fig. 3 shows the successive states of the automata lattice. Each line shows the tape's state in each iteration. We have assigned a different grey level to each cluster shown in Fig. 2: black corresponds to the cluster represented with circles, dark grey to the triangles, light grey to the stars and white to the squares. In the initial iterations, which are shown in the left side, the data items keep the initial random order. According the algorithm operates, the data items...
begin to keep together and the clusters start to appear. Around the 75 iteration – approximately in the middle of the left diagram – the black and greys clusters are almost clustered, just remain some data items belonged to these clusters in the right side. The white cluster is divided into two noncontiguous clusters. Around the 200 iteration the black cluster is completely formed and keeps in such state until the end. The final tape’s state, in which the data items are grouped into the four existing clusters, is achieved in the iteration 283. Once the convergence is reached, this state is maintained. In this case the items are classified with no error.

Straight afterwards the post-processing step oriented to the automatic grouping of the data items scattered on the tape is applied. As the data items are linearly grouped in the cellular automaton tape a straightforward way of finding the natural clusters within the data is by analyzing the chainmap diagram formed by the distances of the successive data items. The main problem is to detect automatically the optimum threshold that gives the correct number of clusters for each dataset.

Fig. 4 shows the chainmap of the synthetic dataset in which the existence of four local maxima, each of one corresponding to an individual cluster can be noticed. These local maxima are located in the 20, 40, 60 and 80 bins. Notice that the last one is needed for defining the frontier between the last cluster and the first one due to we are considering periodic boundary conditions.

Fig. 5 depicts the index $J$ defined in (4) for several number of clusters. We are displaying the values for $J$ (vertical axis) up to 40 clusters. It can be noticed that the minimum is reached at four clusters.

We have also employed the standard $k$-means clustering with this synthetic dataset. The $k$-means clustering algorithm requires the number of classes as an input parameter. For $k=4$ similar successful results can be achieved. As in our proposed algorithm we have used the Euclidean distance as metric.

4.2. Clustering results with the Iris dataset

We have also tested the one-dimensional cellular automata-based clustering algorithm with a real dataset as the Iris dataset. The Iris dataset [15] was first used and even created by Fisher [16] in his pioneering research work on linear discriminant analysis, and today it is still an up-to-date, standard pattern recognition problem for testing discriminant techniques and algorithms.

In this well-known and classical multiclass pattern recognition problem, three classes of Iris flowers (setosa, versicolor and virginica) have to be classified according to four continuous discriminant variables measured in centimeters: sepal length, sepal width, petal length and petal width.

Fig. 6 shows the three classes. We have only represented three variables of this four-dimensional dataset: the sepal length, the sepal width and the petal width. The triangle marks represent the Iris setosa, the circles are the Iris versicolor items and the squares correspond to the Iris virginica.

It is well known that this dataset only contains two clusters with an obvious separation. The Iris setosa is in one of those clusters, while the other two species, Iris versicolor and Iris virginica, are in the other cluster.

Fig. 7 shows the successive states of the automata lattice. As in the synthetic case each line represents the tape’s line in each iteration. As the Iris dataset contains three classes we have only employed the black, grey and white colors to represent the Iris setosa, Iris versicolor and Iris virginica data items, respectively.

The final tape’s state is achieved in the iteration 1291 although a perfect clustering is not obtained in this case. Only the black zone which represents the Iris setosa shows a compact state. Some individuals corresponding to the Iris versicolor (grey...
cluster) are included in the middle of the Iris virginica zone (white cluster) and vice versa.

Fig. 8 shows the chainmap of the Iris dataset. In this case three local maxima can be clearly distinguished, each of one corresponding to an individual cluster.

Fig. 9 shows the index \( J \) defined in (4) for several number of clusters. The values of \( J \) are shown in the vertical axis, while the horizontal one shows the number of clusters considered (from 1 to 40) axis, up to 40 clusters. It can be noticed that the minimum value is reached at three clusters which coincide with the number of clusters in the dataset.

As commented above, the cellular automata clustering has been not able to classify all the items correctly. It is well known that the Iris versicolor and Iris virginica clusters are very hard to separate. This fact can be clearly observed in the matching matrix shown in Table 1. All the 50 Iris setosa items are classified in the correct cluster, three Iris versicolor items are erroneously classified as Iris virginica, and one Iris virginica item is erroneously classified as Iris versicolor.

### Table 1

<table>
<thead>
<tr>
<th>Iris dataset matching matrix obtained with the proposed algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris setosa</td>
</tr>
<tr>
<td>Iris setosa</td>
</tr>
<tr>
<td>Iris versicolor</td>
</tr>
<tr>
<td>Iris virginica</td>
</tr>
</tbody>
</table>
methods such as the data items. This method uses an index based on the intra-cluster of the chainmap diagrams formed by the distances of the successive automatic method for determining this number based on the analysis ordered in the one-dimensional discrete lattice, we have defined an number of clusters in the dataset. A st ed a t ai t e m sa r ec o r r e c t l y and also Ant Clustering algorithms.

5. Conclusions and further work

A novel algorithm for data clustering based on linear cellular automata has been proposed. The method identifies the individual data items as cells belonging to a uni-dimensional cellular automaton and it is inspired in both social segregation models and also Ant Clustering algorithms.

Moreover we introduce a post-processing stage for finding the number of clusters in the dataset. As the data items are correctly ordered in the one-dimensional discrete lattice, we have defined an automatic method for determining this number based on the analysis of the chainmap diagrams formed by the distances of the successive data items. This method uses an index based on the intra-cluster variances, which always tries to obtain the minor number of clusters.

The results obtained as synthetic as real datasets improve significantly the ones obtained with conventional unsupervised methods such as the \(k\)-means algorithm.

We are still applying the proposed cellular automata clustering algorithm to other real datasets where we are obtaining quite similar results. Currently we are optimizing the global performance of the clustering algorithm. When it is used with larger or high-dimensional datasets, the computing time could be a possible drawback due to the number to rules to be applied in each iteration step is increased.

References


Javier de Lope (SM’94, MG’98) received the MSc degree in Computer Science from the Universidad Politécnica de Madrid in 1994 and the PhD degree at the same university in 1998. Currently, he is a Associate Professor in the Department of Applied Intelligent Systems at the Universidad Politécnica de Madrid. His current research interest is centered on the study, design and construction of modular robots and multi-robot systems, and in the development of control systems based on soft-computing techniques. He is currently leading a three-year R&D project for developing industrial robotics mechanisms which follow the guidelines of multi-robot systems and reconfigurable robotics. In the past he also worked on projects related to the computer-aided automatic driving by means of external cameras and range sensors and the design and control of humanoid and flying robots.

Dario Maravall (IEEE SM’78, IEEE M’80) was born in Salamanca (1952) received the MSc in Telecommunications Engineering from the Universidad Politécnica de Madrid in 1978 and the PhD degree at the same university in 1980. From 1980 to 1986, he was a Associate Professor at the School of Telecommunications Engineering, Universidad Politécnica de Madrid. In 1988 he was promoted to Full Professor at the Faculty of Computer Science, Universidad Politécnica de Madrid. His current research interests include computer vision, autonomous robots and computational intelligence. He has published extensively on these subjects and has directed more than 20 funded projects, including a five-year R&D project for the automated inspection of wooden pallets using computer vision techniques and robotic mechanisms, with several operating plants in a number of European countries (Spain, France, Italy and United Kingdom) and in USA (video). As a result of this project he holds a patent issued by the European Patent Office at The Hague, The Netherlands.